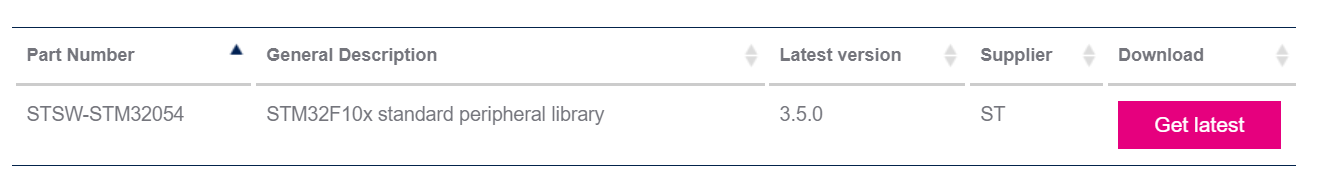
# Platform

Proteus\_8.9\_SP2 (stm32f103c6)

Keil MDK5

# 1. download STM32F103C6 library (version 3.5.0)

https://www.st.com/content/st\_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32-standard-peripheral-libraries/stsw-stm32054.html



# 2. copy file to project

new BOOT, FreeRTOS, Interrupt, Library, Src directory in project directory. copy file

1. copy to boot directory

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\CoreSupport\core\_cm3.c

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\CoreSupport\core\_cm3.h

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm\startup\_stm32f10x\_ld.s

2.copy to library directory

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\STM32F10x\_StdPeriph\_Driver\

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\stm32f10x.h

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\system\_stm32f10x.c

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\system\_stm32f10x.h

3.copy to interrupt diectory

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Examples\GPIO\IOToggle\stm32f10x\_it.c

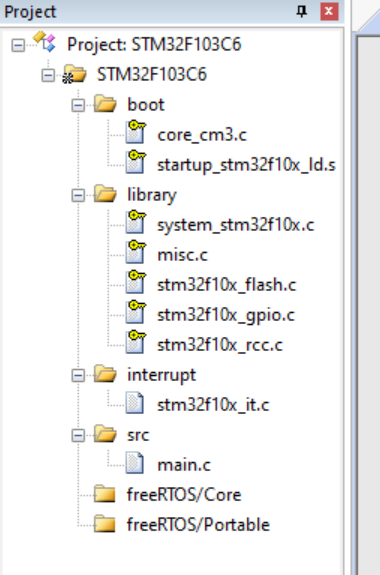
STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Examples\GPIO\IOToggle\stm32f10x\_it.h

4.copy to src directory

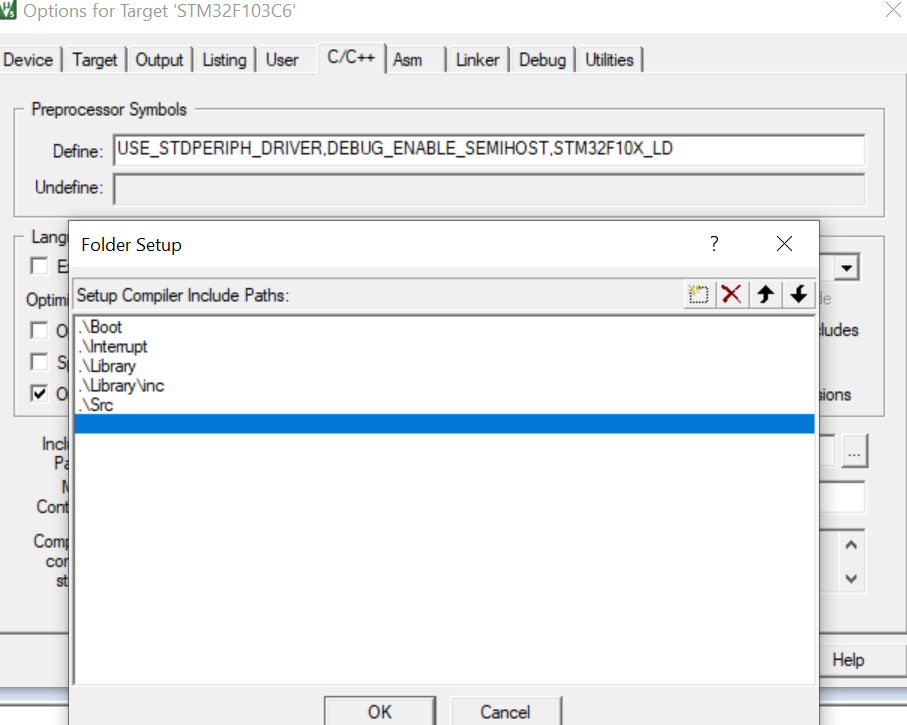
STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Examples\GPIO\IOToggle\main.c

STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Examples\GPIO\IOToggle\stm32f10x\_conf.h

add file to project

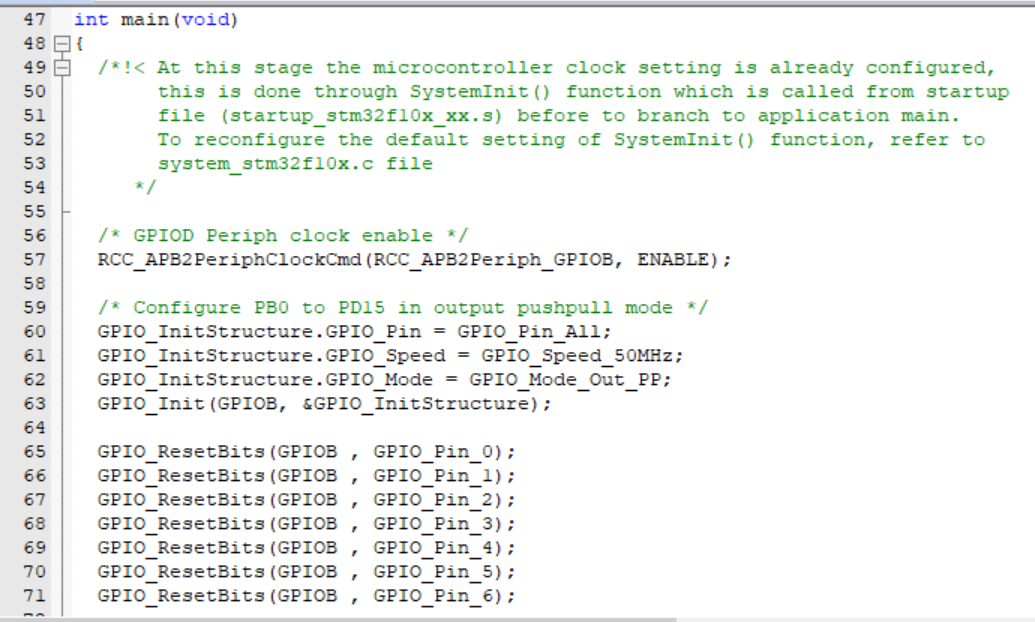


define and config

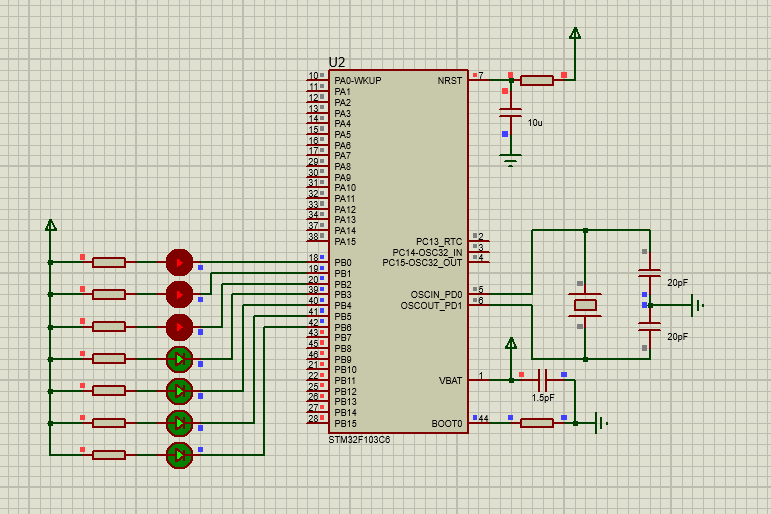


# 3. modify and test

修改Main文件



测试结果



# 4. porting FreeRTOS

1. download freeRTOS

https://www.freertos.org/a00104.html

2. new core, include, portable directory in project directory and porting code

copy flowing code from FreeRTOS to project/FreeRTOS/core

FreeRTOSv202104.00\FreeRTOS\Source\croutine.c

FreeRTOSv202104.00\FreeRTOS\Source\event\_groups.c

FreeRTOSv202104.00\FreeRTOS\Source\list.c

FreeRTOSv202104.00\FreeRTOS\Source\queue.c

FreeRTOSv202104.00\FreeRTOS\Source\stream\_buffer.c

FreeRTOSv202104.00\FreeRTOS\Source\tasks.c

FreeRTOSv202104.00\FreeRTOS\Source\timers.c

copy include file from FreeRTOS source code to project/FreeRTOS/include

FreeRTOSv202104.00\FreeRTOS\Source\include

copy to project/FreeRTOS/portable

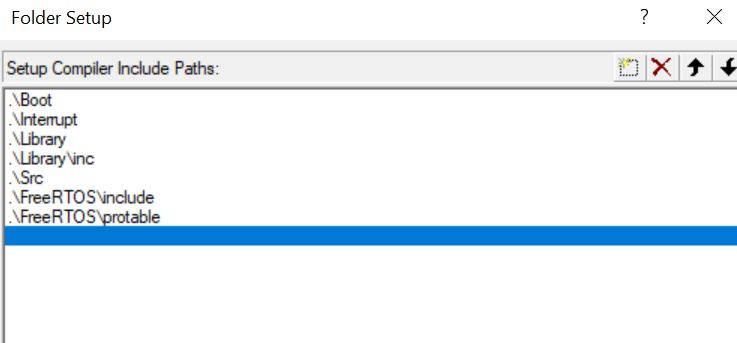
FreeRTOSv202104.00\FreeRTOS\Source\portable\MemMang\

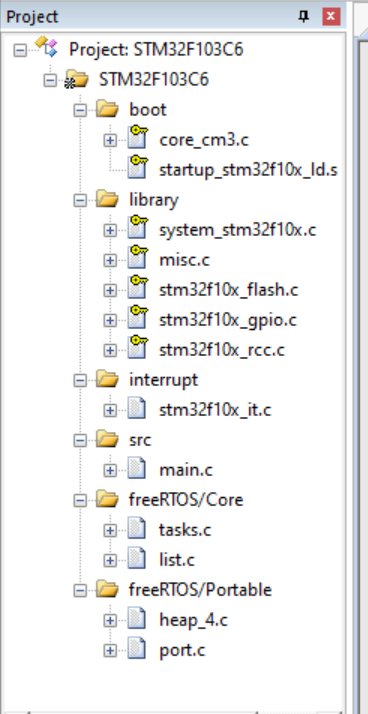
reeRTOSv202104.00\FreeRTOS\Source\portable\RVDS\ARM\_CM3\port.c

reeRTOSv202104.00\FreeRTOS\Source\portable\RVDS\ARM\_CM3\portmacro.h

FreeRTOSv202104.00\FreeRTOS\Demo\CORTEX\_STM32F103\_Keil\FreeRTOSConfig.h

add include file directory





modify code

add include file in main,c

#include "FreeRTOS.h"

#include "task.h"

add in main.c

#define START\_TASK\_PRIO 1

#define START\_STK\_SIZE 64

TaskHandle\_t StartTask\_Handler;

void start\_task(void \*pvParameters);

#define LED0\_TASK\_PRIO 2

#define LED0\_STK\_SIZE 30

TaskHandle\_t LED0Task\_Handler;

void led0\_task(void \*pvParameters);

#define LED1\_TASK\_PRIO 2

#define LED1\_STK\_SIZE 30

TaskHandle\_t LED1Task\_Handler;

void led1\_task(void \*pvParameters);

add in void main() function

xTaskCreate((TaskFunction\_t )start\_task,

(const char\* )"start\_task",

(uint16\_t )START\_STK\_SIZE,

(void\* )NULL,

(UBaseType\_t )START\_TASK\_PRIO,

(TaskHandle\_t\* )&StartTask\_Handler);

vTaskStartScheduler();

while (1)

{

;

}

add in main.c

void start\_task(void \*pvParameters)

{

taskENTER\_CRITICAL();

xTaskCreate((TaskFunction\_t )led0\_task,

(const char\* )"led0\_task",

(uint16\_t )LED0\_STK\_SIZE,

(void\* )NULL,

(UBaseType\_t )LED0\_TASK\_PRIO,

(TaskHandle\_t\* )&LED0Task\_Handler);

xTaskCreate((TaskFunction\_t )led1\_task,

(const char\* )"led1\_task",

(uint16\_t )LED1\_STK\_SIZE,

(void\* )NULL,

(UBaseType\_t )LED1\_TASK\_PRIO,

(TaskHandle\_t\* )&LED1Task\_Handler);

vTaskDelete(StartTask\_Handler);

taskEXIT\_CRITICAL();

}

void led0\_task(void \*pvParameters)

{

while(1)

{

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_0);

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_1);

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_2);

vTaskDelay(10);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_0);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_1);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_2);

vTaskDelay(10);

}

}

void led1\_task(void \*pvParameters)

{

while(1)

{

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_3);

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_4);

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_5);

GPIO\_ResetBits(GPIOB , GPIO\_Pin\_6);

vTaskDelay(10);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_3);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_4);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_5);

GPIO\_SetBits(GPIOB , GPIO\_Pin\_6);

vTaskDelay(20);

}

}

修改FreeRTOSConfig.h

//#define configTOTAL\_HEAP\_SIZE ( ( size\_t ) ( 17 \* 1024 ) )

#define configTOTAL\_HEAP\_SIZE ( ( size\_t ) ( 5 \* 1024 ) )

#define xPortPendSVHandler PendSV\_Handler

#define xPortSysTickHandler SysTick\_Handler

#define vPortSVCHandler SVC\_Handler

修改stm32f10x\_it.c

comment

//void SysTick\_Handler(void)

//{

//}

//void PendSV\_Handler(void)

//{

//}

//void SVC\_Handler(void)

//{

//}

add NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_4); in void main()

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_4);